

# Aplikasi Kurir Mobil Pick Up Berbasis mutli Platform (Multi Platform Pick Up Car Courier Application)

Novaely Rachman<sup>[1]</sup>, Pahrul Irfan<sup>[2]\*</sup>

<sup>[1],[2]</sup>Ilmu Komputer, Universitas Bumigora

E-mail: novaely.r@gmail.com, irfan@universitasbumigora.ac.id

## KEYWORDS:

Courier Platform Android  
Multiplatform  
REST Web  
Service

## ABSTRACT

*Problem faced by goods delivery service providers today is that there are found consumers they have difficulty in finding couriers to transport large goods, thus causing consumers to have to go to the owner of a pickup car or delivery service provider, it can take a lot of time and cost. To solve the problem, there is currently no system that facilitates consumers in ordering couriers optimally. Therefore, it requires a delivery system that can order couriers only through a smartphone owned. The result that was wanted to be achieved in this study was the creation of a multiplatform based courier ordering system with a REST web service architecture that can be accessed through computer and Android platforms. With the construction of this system testing of the web service, the test results of the web service function stated that, all functions tested show valid results, this means that the entire web service function on the application, already fulfilled the design, in addition to testing against the web service, conducted also the black box trial, based on the black box test results, it is concluded that the application created, already according to the design, is proven by all the parts tested showing valid status*

## KATA KUNCI:

Kurir Platform Android  
Multiplatform  
REST Web Service

## ABSTRAK

*Permasalahan yang dihadapi oleh penyedia jasa pengiriman barang saat ini adalah ada ditemukan konsumen mereka mengalami kesulitan dalam mencari kurir untuk mengangkut barang yang berukuran besar, sehingga menyebabkan konsumen harus mendatangi sendiri pemilik mobil pick up atau penyedia jasa pengiriman barang, hal tersebut dapat memakan banyak waktu dan biaya. Untuk menyelesaikan permasalahan tersebut, saat ini belum ada ditemukan sistem yang memudahkan konsumen dalam pemesanan kurir secara optimal. Oleh karena itu dibutuhkan sebuah sistem pengiriman yang dapat memesan kurir hanya melalui smartphone yang dimiliki. Hasil yang ingin dicapai dalam penelitian ini adalah terciptanya sebuah sistem pemesanan kurir berbasis multiplatform dengan arsitektur REST web service yang dapat diakses melalui platform komputer dan Android. Dengan dibangunnya sistem ini dilakukan pengujian terhadap web service, hasil uji coba terhadap fungsi web service dinyatakan bahwa, semua fungsi yang diuji coba menunjukkan hasil valid, hal ini berarti seluruh fungsi web service pada aplikasi, sudah memenuhi rancangan, selain pengujian terhadap web service, dilakukan juga uji coba blackbox, berdasarkan hasil uji coba blackbox, disimpulkan bahwa aplikasi yang dibuat, sudah sesuai dengan rancangan, dibuktikan dengan semua bagian yang diuji coba menunjukkan status valid.*

## I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi saat ini semakin maju dan berkembang. Teknologi informasi memiliki peran penting dalam perkembangan peradaban manusia tanpa adanya teknologi manusia tidak akan sampai sekarang ini. Siapa yang menguasai informasi maka ia yang memiliki peluang lebih dibandingkan yang tidak memiliki. Pemanfaatan informasi yang optimal dapat memberikan ide yang inovatif untuk pengembangan.

Mataram merupakan salah satu kota yang berada di Provinsi Nusa Tenggara Barat tepatnya di Pulau Lombok. Bidang ekonomi, pariwisata, pendidikan masih menjadi faktor utama yang menyebabkan banyak pendatang berkunjung ke kota Mataram. Bahkan tidak terkecuali beberapa pendatang dari luar pulau. Berbagai macam profesi juga terdapat di kota Mataram salah satunya adalah penyedia jasa kurir. Menurut[1] Kurir adalah utusan yang menyampaikan sesuatu yang penting dengan cepat. Kurir sangat dibutuhkan dalam pengantaran barang dan kebutuhan pokok lainnya sampai dengan

selamat ke tujuan. Jasa pengiriman ini telah diimplementasikan sejak zaman dahulu kala dimana para pedagang memanfaatkan kurir untuk memenuhi suplai barang dagangan mereka, hal ini terus berkembang sampai dengan sekarang dimana banyak sekali penyedia jasa layanan pengiriman barang antar kota, pulau, bahkan Negara.

Pada umumnya para penyewa jasa kurir kilat yang ada di kota Mataram memesan jasa melalui aplikasi Grab, Go-Jek, M-Jek dan sebagainya. Akan tetapi kekurangan dari aplikasi ini tidak menyediakan layanan untuk mengantar barang yang banyak dan berukuran besar.

FastCourier adalah badan Startup yang menyediakan jasa layanan kurir yang ada di kota Mataram. FastCourier sudah lumayan banyak dikenal masyarakat Mataram, akan tetapi Startup ini masih belum memiliki aplikasi dan masih mengandalkan pemesanan melalui aplikasi WhatsApp, SMS, dan Instagram.

Dikarenakan hal tersebut penulis mencoba membangun sebuah aplikasi untuk Startup Fast Courier yang mana akan memudahkan konsumen dalam memesan kurir. Kurir disini adalah kurir yang menggunakan mobil pickup sehingga dapat mengangkut banyak barang. Aplikasi ini dapat digunakan untuk membantu dalam pindahan rumah, mengangkut pasir, batu bata, dan barang-barang berat lainnya. Diharapkan aplikasi ini dapat memudahkan masyarakat yang ada di kota Mataram dalam pemesanan jasa kurir serta menambah mata pencarian penduduk yang ada di kota Mataram sendiri. Untuk perancangan dan pembuatan aplikasi ini penulis menggunakan metodologi Waterfall, yaitu metode mengembangkan perangkat lunak yang diawali dengan menganalisa kebutuhan perusahaan, merancang media, membangun aplikasi dengan tools, dan mengujicobanya.

Hasil yang ingin dicapai oleh penulis adalah agar dengan menggunakan REST Web Service ini konsumen atau penyewa jasa layanan kurir dari Startup FastCourier dapat memesan kurir secara optimal tanpa banyak memakan waktu dan biaya.

## II. METODOLOGI

### A. Waterfall

*Waterfall* merupakan metode pengembangan perangkat lunak yang sistematis dan sekuensial yang dimulai pada tingkat kemajuan sistem dan seluruh

analisis, desain, kode (implementasi), pengujian, dan pemeliharaan[2]. Berikut tahapan-tahapan dalam metode *waterfall*:

#### 1) Analisa kebutuhan

Langkah ini merupakan analisa terhadap kebutuhan sistem. Pada tahap ini mendefinisikan tentang semua kebutuhan secara garis besar sistem yang akan di buat.

#### 2) Desain Sistem

Tahapan desain sistem dilakukan untuk menuangkan pikiran dan perancangan sistem terhadap solusi dari permasalahan yang ada dengan menggunakan perangkat pemodelan sistem seperti relasi tabel, *Class Diagram*, *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, Desain *Interface* dan Desain API.

#### 3) Implementasi

Implementasi adalah tahap penerjemahan desain sistem ke dalam bahasa yang dapat di kenali oleh komputer. Pada tahap inilah akan di bentuk aplikasi yang dapat digunakan secara nyata.

#### 4) Pengujian

Pengujian merupakan tahap akhir dimana sistem yang baru akan diuji kemampuan dan keefektifannya dalam melakukan manipulasi data. Proses pengujian dilakukan dengan metode blackbox. Blackbox adalah proses menguji kesesuaian input yang di berikan dengan output yang di harapkan. Metode ini melibatkan pengguna dengan kuesioner sebagai alat ukur untuk menilai apakah sistem yang baru sudah dapat memenuhi kebutuhan pengguna atau belum[3].

### B. Analisis Data dan Informasi

Sebelum melakukan perancangan sistem baru, diperlukan informasi yang jelas mengenai sistem lama yang sedang diterapkan, untuk memperoleh gambaran umum mengenai prosedur kerja sistem lama. Hal ini dilakukan untuk mengetahui bagian mana dari prosedur tersebut yang perlu ditingkatkan atau diganti untuk menghasilkan sistem yang dapat berjalan pada *multiplatform*.

### C. Identifikasi Masalah

Berdasarkan hasil analisa yang telah dilakukan, ditemukan beberapa hal yang dianggap sebagai kendala dari sistem lama, antara lain:

#### 1) Kecepatan Pencarian

Seorang pemesan jasa pengantaran barang mengalami kesulitan dalam pencarian jasa pengantaran untuk mengangkut barang yang berat

dan besar. Para pemesan harus mencari dan menyewa mobil pickup.

## 2) Aplikasi yang dapat membantu

Tidak terdapat sistem yang dapat membantu pemesanan jasa pengantaran barang yang berat.

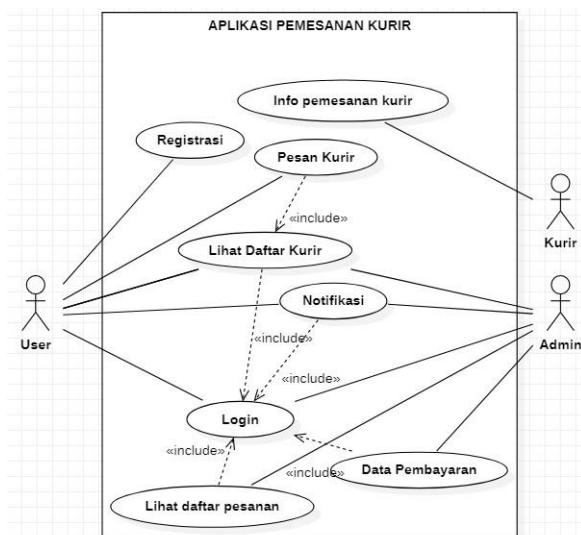
## 3) Pemilihan platform pengguna

Pengguna yang tidak diketahui secara pasti *device*/jenis perangkat yang digunakan dalam pengembangan sistem. Maka diperlukan sistem yang dapat dikembangkan pada *multiplatform* (lebih dari satu *platform*).

## D. Tahap system and software design

### 1) Use Case Diagram

*Use case* digunakan untuk menggambarkan keterkaitan antara user dengan aplikasi yang dibangun[4]. Dalam perancangan sistem ini peneliti menggambarkan *use case* diagram untuk melihat fitur fitur yang digunakan oleh setiap user, desain *use case* diagram sistem dapat dilihat pada Gambar 1:



Gbr 1. Use Case diagram

Penjelasan desain *Use Case* diagram:

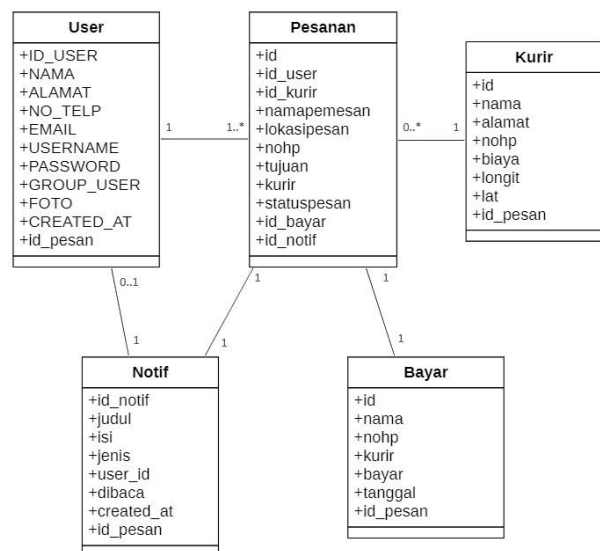
- Aktor yang berperan disini adalah User dan Admin dan Kurir.
- User harus melakukan registrasi dulu pada aplikasi agar dapat login ke dalam aplikasi.
- Admin sudah terdaftar otomatis didalam didalam aplikasi.
- Aktor memiliki *use case* berbeda, relasi *include* menjelaskan bahwa, jika aktor akan menjalankan *use case* tertentu maka harus melewati *use case* sebelumnya.
- Disini User hanya dapat melihat daftar saat

sedang melakukan pemesanan kurir.

- Aktor hanya dapat melakukan pemesanan setelah memilih salah satu dari kurir yang tersedia.
- Admin bisa melihat daftar pesanan yang dimasukkan oleh User dan meneruskan pesanan tersebut ke kurir.
- Admin memberikan notifikasi ke user apakah kurir mau menerima pesanan atau tidak.
- Admin mengisi data transaksi yang dilakukan.
- Kurir hanya mendapat informasi pemesanan kurir.

### 2) Class Diagram

Dalam perancangan sistem ini peneliti menggambarkan class diagram untuk melihat identitas pada setiap class, desain rancangan class diagram dapat dilihat pada gambar 2 dibawah ini:



Gbr 2. Class Diagram

Penjelasan desain *class* diagram:

- Class* pesanan merupakan objek yang menggambarkan kegiatan pemesanan antara konsumen dan admin.
- Class* user terhubung dengan class pemesanan dan notif, karena hanya user saja yang bisa melakukan proses pemesanan dan mendapatkan notifikasi pemesanan.
- Class* kurir hanya terhubung dengan class pemesanan karena disini class kurir hanya sebagai sumber data untuk pemesanan dan bukan sebagai pengguna aplikasi.

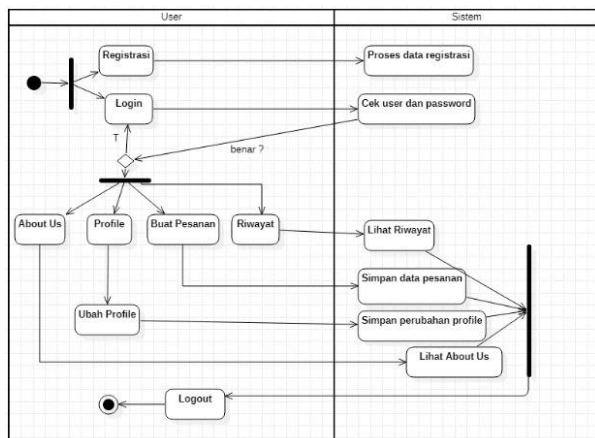
- d) *Class* notif terhubung dengan class user, menggambarkan hanya user saja yang bisa mendapatkan notifikasi pesanan.
- e) *Class* bayar untuk menyimpan semua transaksi pembayaran yang terjadi.

### 3) Activity Diagram

*Activity diagram* menggambarkan aktifitas-aktifitas, objek, state transisi state dan event. Dengan kata lain activity diagram menggambarkan alur kerja perilaku sistem untuk suatu aktifitas[5].

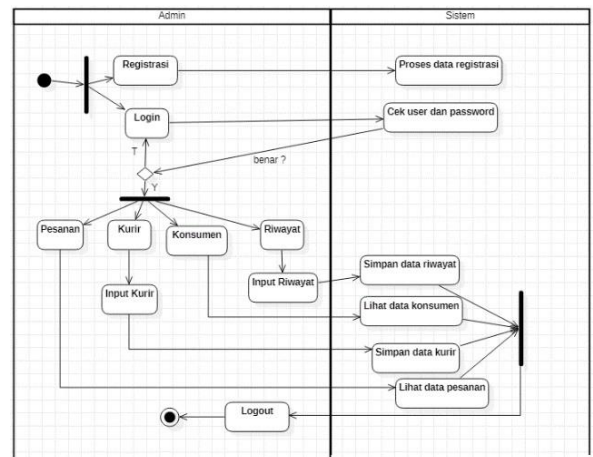
Peneliti menggunakan *activity diagram* untuk menggambarkan urutan aktivitas yang terjadi di dalam aplikasi. Peneliti menggambarkan *activity diagram* terbagi menjadi 2 yaitu *activity diagram* user dan *activity diagram* admin. Berikut adalah *activity* yang peneliti gambarkan beserta penjelasannya:

- a) *Activity diagram* user, merupakan *activity* keseluruhan sistem yang akan dilakukan oleh seorang konsumen atau pemesan layanan kurir. *Activity diagram* dari user dapat dilihat pada gambar 3 berikut ini:



Gbr 3. Activity Diagram User

- b) *Activity diagram* Admin, merupakan *activity* keseluruhan sistem yang akan dilakukan oleh seorang admin. *Activity diagram* admin dapat dilihat pada Gambar 4 dibawah ini:



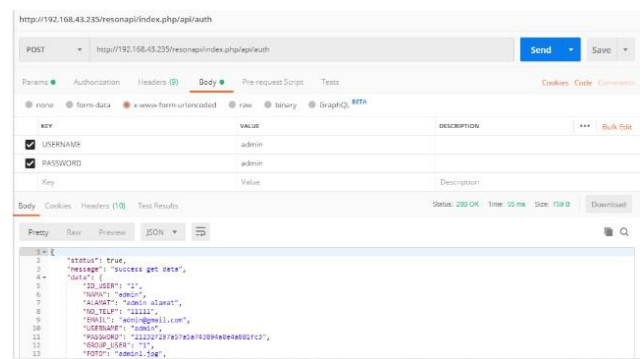
Gbr 4. Activity Diagram admin

## III. HASIL DAN PEMBAHASAN

Tahap implemementasi dimulai dengan pembuatan *database* di *local server* menggunakan *phpmyadmin* yang merupakan salah satu fitur Xampp. dilanjutkan dengan membuat tabel tabel yang dibutuhkan untuk pembuatan sistem. Setelah *database* dibuat, kemudian dilakukan proses pembuatan API Webservice yang digunakan untuk melayani request client terhadap server dan respon server terhadap client yang berupa data JSON (JavaScript Object Notation) yang dapat diolah kembali oleh client. Bahasa pemrograman yang digunakan penulis untuk membangun API Web service ini adalah PHP dan framework codeigniter serta *database* yang digunakan adalah mysql. Setelah API Web service selesai dibuat barulah dilakukan pembuatan tampilan antarmuka aplikasi berbasis *hybrid* menggunakan bahasa pemrograman *Javascript* dengan framework *Phonegap Cordova*. Berikut adalah tampilan pengujian API yang dibangun:

### 1) Endpoint API

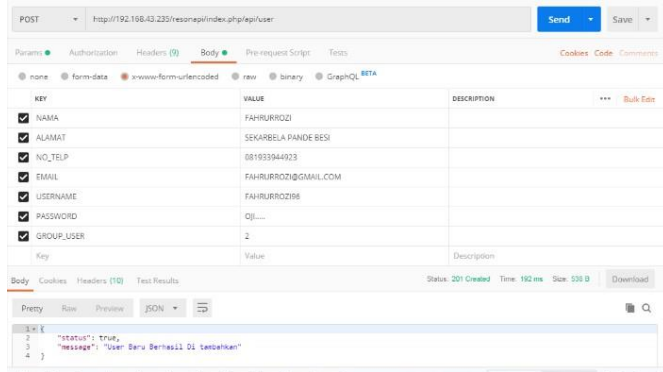
- a) URI /api/auth (POST)



Gbr 5. API Auth

Pada gambar 5 diatas, kita bisa melihat bahwa : apabila kita mengirim data berupa username = admin, dan password = admin pada URL/api/auth dengan method POST maka akan mendapat tanggapan berupa data dari user yang memiliki username admin.

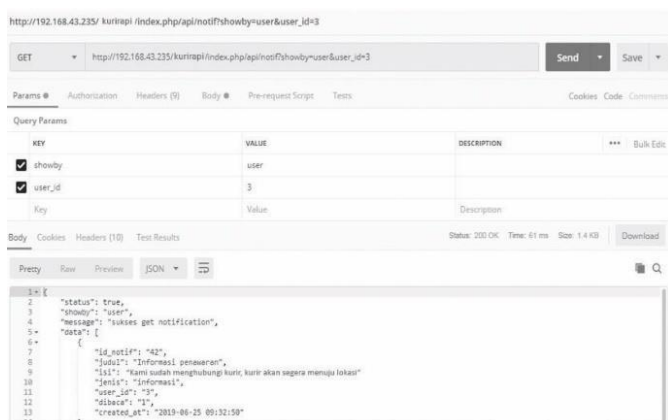
**b) URI /api/user (POST)**



**Gbr 6. User POST**

Pada gambar 6 diatas, kita bisa melihat bahwa : apabila kita mengirim data nama, alamat, no\_telp, email, username, password, group\_user pada URL/api/user dengan method POST maka akan mendapat tanggapan berupa “User baru Berhasil Di Tambahkan”, yang artinya data user baru sudah masuk ke dalam database aplikasi.

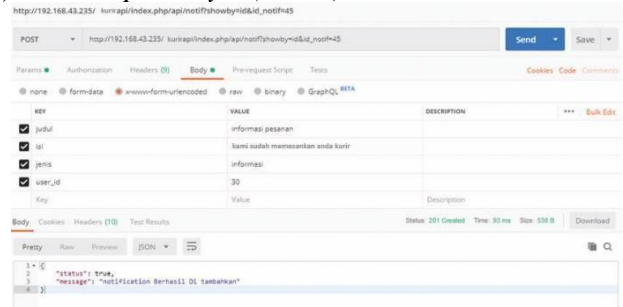
**c) URI /api/riwayat (GET)**



**Gbr 7. User GET**

Pada gambar 7 diatas, kita bisa melihat bahwa : apabila kita melakukan request GET pada url api/riwayat, maka akan mendapatkan tanggapan berupa data riwayat-riwayat yang ada pada database

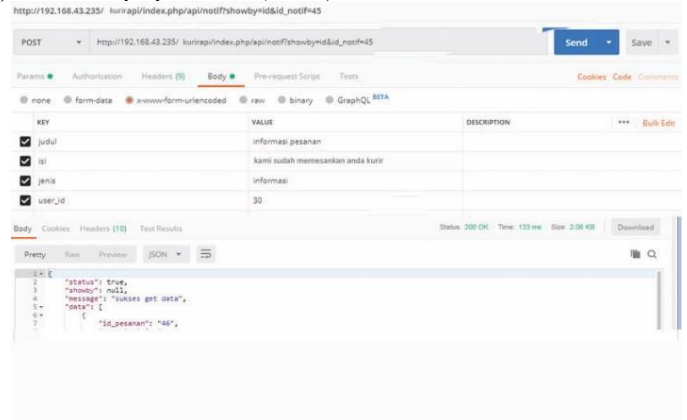
**d) URI /api/riwayat (POST)**



**Gbr 8. API Riwayat**

Pada gambar 8 diatas, kita bisa melihat bahwa : apabila kita mengirim data judul, isi, jenis, user\_id pada URL/api/riwayat dengan method POST, maka akan mendapat tanggapan riwayat berhasil di tambahkan, yang artinya data riwayat yang dikirim sudah masuk kedalam database.

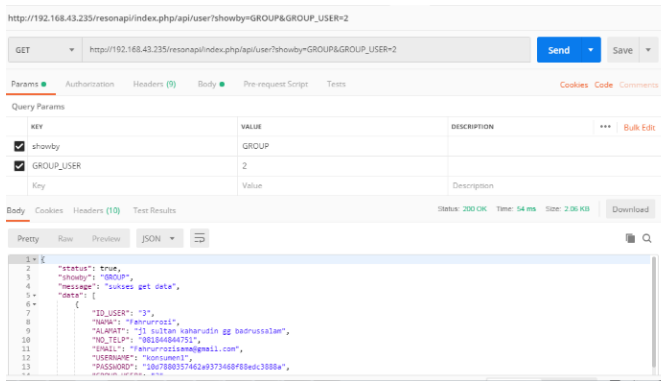
**e) URI /api/pesanan (GET)**



**Gbr 9. API Pesanan**

Pada gambar 9 diatas, kita bisa melihat bahwa : apabila kita melakukan request GET pada URL api/pesanan, maka akan mendapatkan tanggapan berupa data pesanan-pesanan yang ada pada database. yang artinya data Pesanan yang ada pada database aplikasi sudah diperbaharui dengan data Pesanan yang baru.

**f) URI /api/user (GET), output /api/user dapat dilihat pada gambar dibawah ini :**



Gbr 10. API get User

Pada gambar 10 diatas, kita bisa melihat bahwa : apabila kita melakukan request GET pada url api/user, maka akan mendapatkan tanggapan berupa data user-user yang ada pada database.

## 2) Interface aplikasi

Aplikasi yang dibangun dapat berjalan pada platform android dan web. Pada penjelasan yang akan dijabarkan dibawah ini hanya pada platform mobile menggunakan system operasi android. Berikut adalah tampilan aplikasi yang dibangun:

### a) Pendaftaran User

Silahkan masukan data anda

Masukan Nama

Masukan No Telepon / WA

Masukan email

Masukan username

Masukan password

Data Alamat

Masukan Kecamatan

Masukan Kelurahan

Masukan Alamat

DAFTAR

Gbr 11. registrasi

Gambar 11 merupakan form registrasi yang harus di isi oleh konsumen apabila ingin mendaftarkan diri pada aplikasi , adapun data yang harus di isi oleh user untuk mendaftarkan diri adalah : Nama, Alamat, No Telepon, Email, Username dan Password,

### b) Login user

FORM LOGIN

Username

Password

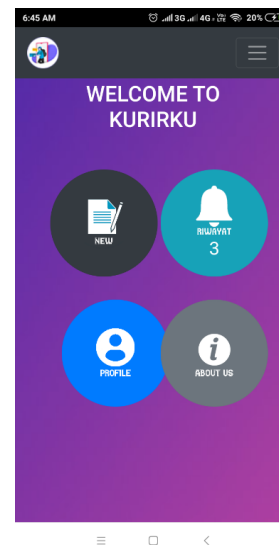
Login

Belum punya akun ? [Daftar Di Sini](#)

Gbr 12. login

Gambar 12 merupakan form login yang harus di isi oleh user apabila ingin masuk kedalam menu awal user pada aplikasi, adapun data yang harus di isi oleh user untuk login adalah : Username dan Password

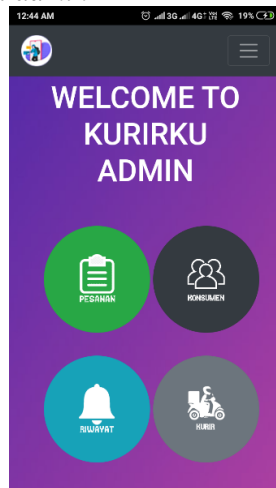
### c) Halaman awal user



Gbr 13. Class Diagram

Gambar 4.9 merupakan halaman awal user yang tampil setelah user dengan kategori group\_user 2 atau konsumen melewati form login, pada halaman ini akan ditampilkan menu bar, serta shortcut untuk menuju ke menu yang lainnya



d) *Halaman awal admin*

Gbr 14. Class Diagram

Pada halaman awal admin terdapat menu pesanan untuk melihat pesanan yang masuk dari konsumen, halaman konsumen, halaman Riwayat dan halaman kurir.

e) *Halaman pesanan*

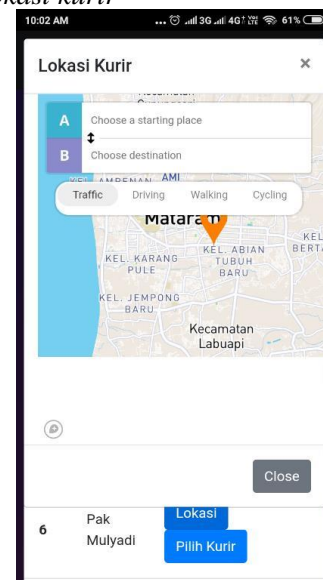
Gbr 15. Class Diagram

Gambar 4.11 Merupakan form tempat user untuk memesan kurir

f) *Halaman data kurir*

Gbr 16. Class Diagram

Pada halaman ini dapat dilihat kurir yang telah mendaftar pada aplikasi.

g) *Halaman lokasi kurir*

Gbr 17. Class Diagram

Pada halaman ini admin dapat melihat lokasi kurir yang mengantarkan barang secara live sehingga admin dapat memantau lokasi kurir yang digunakan.

3) *Pengujian aplikasi*

Pengujian aplikasi menggunakan metode *blackbox* untuk melihat fungsionalitas dari aplikasi apakah dapat berjalan sesuai dengan rancangan. hasil pengujian dengan metode *blackbox* dapat ditunjukkan dengan tabel berikut ini:

TABEL I  
HASIL PENGUJIAN APLIKASI

Aktivitas Pengujian	Yang diharapkan	Pengamatan	Kesimpulan
Registrasi	Input data kurir untuk pendaftaran	Data tersimpan pada aplikasi	Diterima
Login	Masuk ke halaman login	Muncul halaman login	Diterima
Halaman utama	Masuk ke halaman utama	Muncul halaman utama	Diterima
Halaman pemesanan kurir	Dapat memesan kurir untuk pengantaran barang	Proses pemesanan berhasil	Diterima
Halaman data kurir	Dapat menampilkan data kurir yang ada	Data kurir dapat ditampilkan	Diterima
Halaman lokasi kurir	Melihat lokasi kurir secara live	Data live terlihat pada peta.	Diterima
Log out	Keluar dari aplikasi	Berhasil logout	Diterima

Hasil pengujian *block box* diatas juga menunjukkan bahwa secara fungsionalitas, sistem mampu berjalan dengan baik tanpa terjadi kesalahan atau *error* yang tidak diinginkan. Ini menunjukkan bahwa system siap untuk diimplementasikan secara penuh

#### IV. KESIMPULAN

Kesimpulan yang dapat diambil dari penelitian yang dilakukan adalah sebagai berikut:

- a. Aplikasi berhasil dibangun dengan melalui tahapan pengembangan perangkat lunak, mulai

dari analisa kebutuhan sampai dengan tahap pengujian aplikasi.

- b. Penelitian ini menghasilkan bisnis proses yang melibatkan user atau pemesan kurir sebagai pengelola
- c. Uji coba program juga dilakukan dengan menggunakan metode *blackbox* ,berdasarkan hasil uji coba, disimpulkan bahwa aplikasi yang dibuat, sudah sesuai dengan rancangan, dibuktikan dengan semua bagian yang diuji coba menunjukkan status *valid*

#### UCAPAN TERIMA KASIH

Terima kasih yang sebesar-besarnya peneliti ucapkan kepada semua pihak yang telah membantu dalam proses pelaksanaan penelitian sampai dengan penyelesaian tulisan pada paper penelitian ini.

#### REFERENSI

- [1] KBBI, (2016). *Kamus Besar Bahasa Indonesia (KBBI)*. [Online] Available at: <http://kbbi.web.id/pusat>, [Diakses 21 januari 2020]
- [2] Roger S. Pressman. (2012). *Rekayasa Perangkat Lunak (Pendekatan Praktisi) Edisi 7 : Buku 1* (7th ed.). Andi Publisher.
- [3] Kadir, A. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. Andi Offset.
- [4] Kristanto, Andri. (2008). *Perancangan Sistem Informasi dan Aplikasinya* (1st ed.). Gava Media.
- [5] Fowler, M. (2005). *UML Distilled*. Andi Offset.